

Research article

DNA-based watermarks using the DNA-Crypt algorithm

Dominik Heider and Angelika Barnekow*

Address: Department of Experimental Tumorbiology, University of Muenster, Badestr. 9, D-48149 Muenster, Germany

Email: Dominik Heider - dominik.heider@uni-muenster.de; Angelika Barnekow* - barneko@uni-muenster.de

* Corresponding author

Published: 29 May 2007

Received: 26 February 2007

BMC Bioinformatics 2007, 8:176 doi:10.1186/1471-2105-8-176

Accepted: 29 May 2007

This article is available from: <http://www.biomedcentral.com/1471-2105/8/176>

© 2007 Heider and Barnekow; licensee BioMed Central Ltd.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Background: The aim of this paper is to demonstrate the application of watermarks based on DNA sequences to identify the unauthorized use of genetically modified organisms (GMOs) protected by patents. Predicted mutations in the genome can be corrected by the DNA-Crypt program leaving the encrypted information intact. Existing DNA cryptographic and steganographic algorithms use synthetic DNA sequences to store binary information however, although these sequences can be used for authentication, they may change the target DNA sequence when introduced into living organisms.

Results: The DNA-Crypt algorithm and image steganography are based on the same watermark-hiding principle, namely using the least significant base in case of DNA-Crypt and the least significant bit in case of the image steganography. It can be combined with binary encryption algorithms like AES, RSA or Blowfish. DNA-Crypt is able to correct mutations in the target DNA with several mutation correction codes such as the Hamming-code or the WDH-code. Mutations which can occur infrequently may destroy the encrypted information, however an integrated fuzzy controller decides on a set of heuristics based on three input dimensions, and recommends whether or not to use a correction code. These three input dimensions are the length of the sequence, the individual mutation rate and the stability over time, which is represented by the number of generations. *In silico* experiments using the Ypt7 in *Saccharomyces cerevisiae* shows that the DNA watermarks produced by DNA-Crypt do not alter the translation of mRNA into protein.

Conclusion: The program is able to store watermarks in living organisms and can maintain the original information by correcting mutations itself. Pairwise or multiple sequence alignments show that DNA-Crypt produces few mismatches between the sequences similar to all steganographic algorithms.

Background

Sensitive information, especially secret information must be protected against unauthorized access. To achieve this researchers have looked for new cryptographic or steganographic techniques. Existing algorithms encrypt or hide information in binary files, however there are other

media, which can be used. There are several algorithms, which encode information into DNA sequences. Examples are the concepts of Clelland et al., Gehani et al., Leier et al, Wong et al. and Arita et al [1-5]. These techniques can be used for authentication or to store data for long time.

Clelland et al

Inspired by the micro-dots used during the 2nd world war, Clelland et al. developed an extension of this principle [1]. The scientists produced artificial DNA strands, which contained secret messages. A triplet encodes one character or number (Table 1). The Clelland algorithm is a simple substitution cipher which encodes characters into DNA sequences using the following encoding function

- $E : X \rightarrow Y$
- $X \in \{A, B, C,..., Z, 0, 1,..., 9, ".","'"," : ","\sqcup"\}$
- $Y \in \{xyz : x, y, z \in \{A, C, G, T\}\}$

The decoding function is corresponding $D : Y \rightarrow X$.

Now Clelland et al. ligated two primers with the synthesized DNA sequences, a forward and a reverse primer. These ligated sequences were mixed up with dummy strands. Important preconditions are:

- length of dummy strand = length of message DNA with primers
- #copies of each dummy = #copies of message DNA

The receiver must know the decoding function and the primer to decode the message. The primers are used for the polymerase chain reaction and in the last step the amplified DNA sequence has to be sequenced and

decoded. To improve the security one can use dummy strands, which are not random but correspond to words out of a dictionary.

Gehani et al

The original One-Time pad uses the XOR – exclusive or (\oplus). In the case of DNA, the XOR is very impracticable and therefore it is better to use the properties of DNA. Gehani et al. established a DNA One-Time pad by creating word pairs [2]. The first word is the plain text and the second one is the cipher text. After such a block of plain and cipher text, there is a stop codon (Figure 1). The DNA polymerase completes the plain and cipher text.

To encode a message, the plain text is mixed with the DNA sequences. It binds directly to the corresponding complementary sequence. The DNA polymerase creates the cipher text accordingly and the decoding is functionally analogous. The cipher text binds to its complement and the DNA polymerase creates the plain text.

Leier et al

Leier et al. encoded binary information into DNA sequences. A short DNA sequence represents the binary 1_2 , another one represents 0_2 [3]. Further there are another two short DNA sequences, which represent start and end. The fragments have sticky ends and can be ligated (Figure 2). All resulting sequences are like this $s\{0_2|1_2\}e$. The start and end marker have primer sequences on one site for the polymerase chain reaction, which can not be ligated.

Although it seems to be more complicated, it is very similar to the algorithm of Clelland et al. The resulting DNA sequence is mixed with dummy strands and can only be detected and isolated knowing the primer sequences.

Wong et al

Wong et al. developed a steganographic algorithm based on DNA, which is able to store data in living organisms [4]. The data are translated into a DNA sequence which is inserted into a vector. The insert sequence is flanked by two primer sequences which do not exist in the genome yet. This vector is introduced into a cell of a living organism where it coexists and is replicated with the genomic DNA. To extract the data they used a polymerase chain reaction.



Figure 1
DNA One-Time pad. A_i: plain text, B_i: cipher text (and primer for the DNA polymerase), black box: stop Modified from Gehani et al. [2].

Table 1: Clelland code table. Modified from Clelland et al. [1].

character = triplet	character = triplet
A = CGA	U = CTG
B = CCA	V = CCT
C = GTT	W = CCG
D = TTG	X = CTA
E = GGC	Y = AAA
F = GGT	Z = CTT
G = TTT	0 = ACT
H = CGC	1 = ACC
I = ATG	2 = TAG
J = AGT	3 = GCA
K = AAG	4 = GAG
L = TGC	5 = AGA
M = TCC	6 = TTA
N = TCT	7 = ACA
O = GGA	8 = AGG
P = GTG	9 = GCG
Q = AAC	\sqcup = ATA
R = TCA	, = TCG
S = ACG	. = GAT
T = TTC	: = GCT

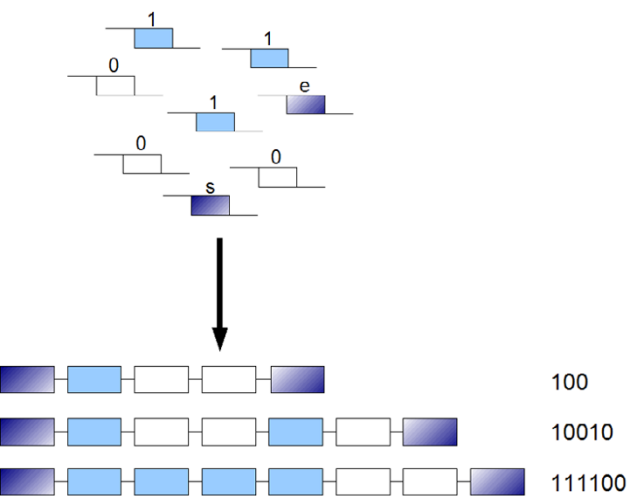


Figure 2
DNA binary strands. Short DNA strands represent the binary 1₂ (light blue), 0₂ (white), start and end marker (dark blue). These sequences can be ligated to long strands by using the sticky ends. Modified from Leier et al. [3].

Wong et al. used a substitution cipher similar to Clelland et al. to encode a song text into a DNA sequence and stored it in *Deinococcus radiodurans*. *Deinococcus radiodurans* survive extreme conditions, e.g. ionizing radiation, so the song text can be stored for hundreds of years.

Arita et al

Arita et al. developed a steganographic algorithm based on the degenerative genetic code. Amino acid codes are redundant so that the translation of mRNA into proteins is a substitution cipher with the following characteristics

- $E : X \rightarrow Y$
- $X \in \{xyz : x, y, z \in \{A, C, G, U\}\}$
- $Y \in \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, STOP\}$

But the inverse function $D : Y \rightarrow X$ is not injective.

An example:

threonine(T) = E(ACU) = E(ACC) = E(ACA) = E(ACG)

The triplet of threonine is redundant in the third base so mutations in the third base do not exert any influence on the translation of threonine and the translated protein. These mutations are called "synonymous substitutions", in contrast to the "non-synonymous substitutions".

Arita et al. translated each letter of the English alphabet into six codons (Table 2). A value of 0 means to keep the original base at the third position of a codon, while a value of 1 means to change the third base at that position. Arita et al. added a parity bit to each letter, to keep it odd for possible error detection [5]. They encoded 'KEIO' into the *ftsZ* gene of *Bacillus subtilis* which is essential for cell division and demonstrated as expected that the changed codon sequences did not affect the cell division, colony morphology, growth rate and sporulation frequency of these bacteria. To extract the encoded message one has to know the original sequence so that one can decide whether the codon is the original or the altered sequence.

Comparison to DNA-Crypt

Clelland et al., Gehani et al. and Leier et al. produced synthesized DNA sequences which were mixed with dummy strands. These sequences contained a secret message. Knowing the unique primer sequence, the secret message can be read out.

Wong et al. and Arita et al. introduced DNA sequences containing a secret message into living organisms. Wong et al. used a vector which incorporated into the genome of *Deinococcus radiodurans* and Arita et al. used point mutations in redundant codons. Arita et al. used a parity bit for error detection. The disadvantage is that if mutations occur, the hidden information is lost.

The DNA-Crypt algorithm is based on small redundant regions comparable to least significant bits in the case of image steganography (Figure 3). The least significant bits encode a difference in colour of just one on the colour scale, not visible to the human eye, and can be used to hide information in images.

Table 2: Arita et al. encryption. The translation table for the English alphabet. Modified from Arita et al. [5].

000001	...	010011	R
000010	E	100011	H
000100	T	001101	D
001000	A	010101	L
010000	O	100101	C
100000	S	011001	M
000111	N	101001	U
001110	F	110100	J
010110	G	111000	Q
011010	W	011111	Z
011100	Y	101111	'
100110	B	110111	.
101010	V	111011	&
101100	K	110001	P
001011	I	111101	
110010	X	111110	

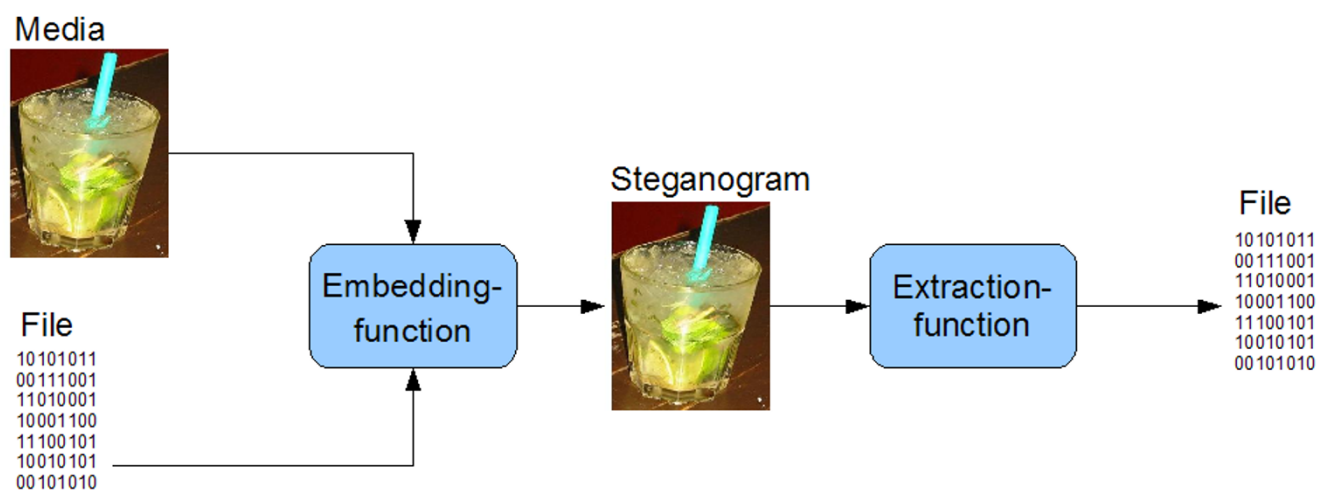


Figure 3
steganographic algorithms. The function of steganographic algorithms.

Text or binary information can also be encoded using any DNA based encryption. However unlike image steganography, the DNA steganography does not lead to a loss of information if the focused range is a protein coding region. DNA-Crypt checks for "synonymous codons" in a genome and point mutations are produced by changing the bases [see Additional file 1].

This algorithm offers the possibility to incorporate data into the genome of living organisms, using an alternative method to Wong et al. [4] (Figure 4). The algorithm is similar to the algorithm of Arita et al., but DNA-Crypt has some important extensions e.g. the use of several encryption and mutation correction codes, which allows encoding of binary information. These extensions are described in the next subsections [5]. A comparative overview of the algorithms and their features is shown in table 3.

Encoding binary information using DNA-Crypt

DNA-Crypt encodes binary information using the following substitution cipher:

- $E : X \rightarrow Y$
- $X = \{xy : x, y \in \{0_2, 1_2\}\}$
- $Y \in \{A, C, G, T\}$

A standard setting is given in table 4.

The binary sequence 0111001001001111_2 would be encoded to $E(0111001001001111_2) = GATCGTAA$.

Two bits could be encoded by one base, so one byte needs four bases for its encoding.

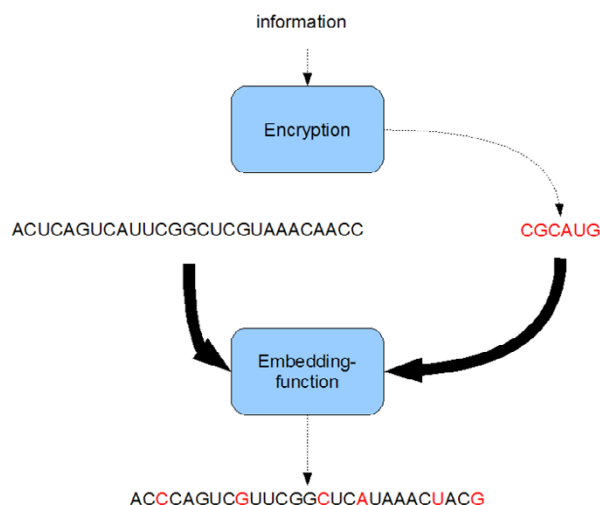


Figure 4
The DNA-Crypt algorithm. The function of the DNA-Crypt algorithm.

Table 3: Comparison of the DNA encryption algorithms

algorithm	organism	affect	error detection	error correction	binary	encryption	utilization
Clelland et al.	-	-	-	-	-	-	20
Gehani et al.	-	-	-	-	-	-	-
Leier et al.	-	-	-	-	+	-	≤ 9
Wong et al.	+	+	+	+	-	-	20
Arita et al.	+	-	+	-	-	-	≤ 5
DNA-Crypt	+	-	+	+	+	+	≤ 8

organism: the use of this algorithm in living organisms;

affect: observation that the algorithm exerts an effect on the organisms;

error detection/correction: the algorithm shows an error detection/correction function;

binary: binary information can be encoded;

encryption: the use of binary encryption algorithm like AES or RSA;

utilization: storage utilization in a 100 bp DNA sequence;

- = negative;

+ = positive;

Based on this binary encryption, several private and public key cryptographic algorithms are integrated in DNA-Crypt:

- One-Time pad [6]
- AES [7]
- Blowfish [6]
- RSA [8,9]

To use DNA-Crypt one has to register so that DNA-Crypt can create AES, Blowfish and RSA keys for the user. These keys can be used to encrypt the binary information which then gets integrated into the genome. In addition it is possible to export and to import these keys and to exchange them with other users. Further the user can create new keys in DNA-Crypt or delete old ones. Another possibility is to use a One-Time pad instead of an encryption key. Compared to Arita et al. our substitution cipher allows to use several encryption algorithms as described above. In addition DNA-Crypt offers a better storage utilization compared to the algorithm of Arita et al. (four instead of six synonymous codons per character).

Table 4: DNA-Crypt binary encryption. The standard settings for encoding binary sequences in DNA-Crypt.

binary sequence	base
00 ₂	T
01 ₂	G
10 ₂	C
11 ₂	A

Mutation correction

Mutations do not occur very often, approximately 10^{-10} to 10^{-15} per cell division, but they can destroy the encrypted information in DNA sequences. To correct these failures DNA-Crypt uses a correction code based on binary correction. One of them is the 8/4 Hamming-code and another one is the WDH-code [10]. The advantage of the WDH-code is that it can correct more mutations than the 8/4 Hamming-code. The n-times WDH-code repeats the encrypted DNA sequence n times. It can correct $\left\lfloor \frac{n-1}{2} \right\rfloor$

failures. All WDH-codes where n is an odd number are perfect.

The 8/4 Hamming-code can only correct ≤ 25% of the mutations. Four bits are used for information (b_3, b_2, b_1, b_0) and the other four bits as parity bits. A complete byte is represented by these eight bits $b_3, b_3 \oplus b_2 \oplus b_1, b_2, \neg b_2 \oplus b_1 \oplus b_0, b_1, \neg b_3 \oplus b_1 \oplus b_0, b_0, \neg b_3 \oplus b_2 \oplus b_0$

which are called $h_7, h_6, h_5, h_4, h_3, h_2, h_1, h_0$. To decode the byte, the following parity sums are build:

$$\bullet p = h_7 \oplus h_6 \oplus h_5 \oplus h_4 \oplus h_3 \oplus h_2 \oplus h_1 \oplus h_0$$

$$c_0 = h_7 \oplus h_5 \oplus h_1 \oplus h_0$$

$$c_1 = h_7 \oplus h_3 \oplus h_2 \oplus h_1$$

$$c_2 = h_5 \oplus h_4 \oplus h_3 \oplus h_1$$

If $p = 1$ there are 0 or 2 failures in the byte. The byte was transmitted correct, if the parity bits c_0, c_1, c_2 are correct, which means equal to 1. If not, there happened 2 failures, which cannot be corrected.

If $p = 0$ there is 1 failure in the byte which can be corrected using table 5.

Only one of four bits can be corrected. But not all mutations can be corrected by the 8/4 Hamming-code. Failures which only differ in one bit can be corrected, e.g. 00 \leftrightarrow 01 or 11 \leftrightarrow 10. Failures like 00 \leftrightarrow 11 or 10 \leftrightarrow 01 cannot be corrected.

The limiting resource for mutation correction is not the time, but the space. The advantage of the 8/4 Hamming-code is that it is very compact. The space requirements of the 8/4 Hamming-code is $f(n) = 2n \in \Theta(n)$. In contrast to

$$f(n) = \begin{cases} n^2, & n \text{ is an odd number} \\ n^2 + n, & \text{else} \end{cases} \in \Theta(n^2) \text{ for the WDH-}$$

code.

For example to encode one byte, which means a DNA sequence of four bases, the 8/4 Hamming-code needs eight synonymous codons instead of twenty synonymous codons for the 5-times WDH-code. In contrast to the data published by Arita et al. we can not only exhibit error detection but error corrections which enables us to maintain the data. This obviously represents an important advantage.

fuzzy controller

The integrated fuzzy controller decides and recommends whether to use the 8/4 Hamming-code, the WDH-code or no mutation correction for optimal performance [11-14] [see Additional file 2]. It uses the Singleton-fuzzyfication and has three input dimensions with each separated into three triangular sets. The first dimension is the individual mutation rate (ϕ) of the DNA sequence containing the secret message (Figure 5). This is based on a standard mutation rate, by default $1 * 10^{-7}$ for prokaryotes and $1 * 10^{-10}$ for eukaryotes, which is changed by specific mutation rates (α_i) for each base pair. These changes are based on the transversion and transition rate and in addition on the stability (δ) of GC rich regions.

Table 5: The 8/4 Hamming-code correction table

c_0	c_1	c_2	failure
1	1	1	h_6
1	1	0	h_4
1	0	1	h_2
0	1	1	h_0
0	0	1	h_7
0	1	0	h_5
1	0	0	h_3
0	0	0	h_1

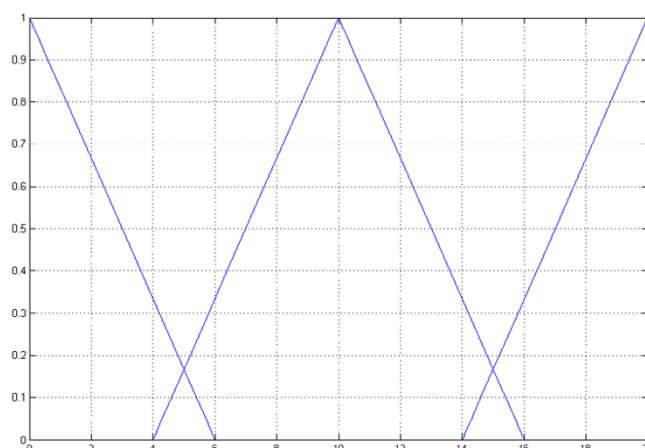


Figure 5

The first input dimension. The first input dimension of the fuzzy controller is the mutation rate. The first input dimension is separated into three triangular sets $X_i = (a_m, a_\lambda, a_\rho)$. The first called "low" = (0, 0, 6) describes a low mutation rate. The second "middle" = (10, 4, 16) and the third "high" = (20, 14, 20) describe a middle and a high mutation rate.

- $\phi = \sum_{i=0}^3 \alpha_i / 4$
- $\alpha_0 = \#C * (\sum_{i=A,G,T}^3 \alpha_{Ci} / 3) / \delta$
- $\alpha_1 = \#G * (\sum_{i=A,C,T}^3 \alpha_{Gi} / 3) / \delta$
- $\alpha_2 = \#A * \sum_{i=C,G,T}^3 \alpha_{Ai} / 3$
- $\alpha_3 = \#T * \sum_{i=A,C,G}^3 \alpha_{Ti} / 3$

The first input dimension is separated into three triangular sets $X_i = (a_m, a_\lambda, a_\rho)$ [15-20]. The first called "low" = (0, 0, 6) describes a low mutation rate. The second "middle" = (10, 4, 16) and the third "high" = (20, 14, 20) describes a middle and a high mutation rate.

The second input dimension is the length of the DNA sequence containing the secret message (Figure 6).

The triangular sets are "short" = (0, 0, 24), "middle" = (40, 16, 64) and "long" = (80, 56, 80).

The third input dimension is the stability over time, which is represented by the number of generations (Figure 7). It is separated into "low" = (0, 0, 400), "middle" = (500, 100, 900) and "high" = (1000, 600, 1000).

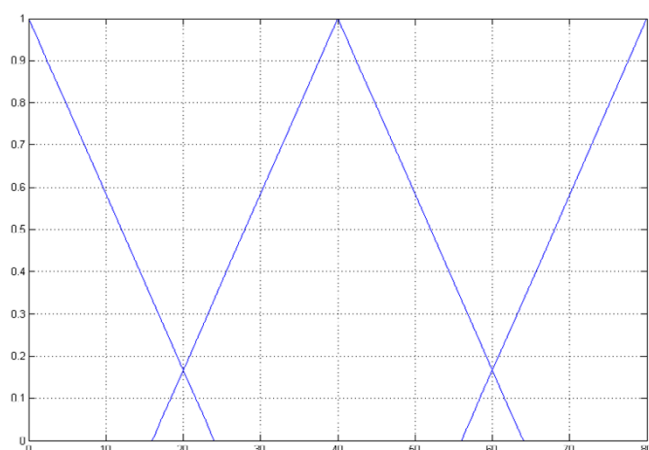


Figure 6

The second input dimension. The second input dimension of the fuzzy controller is the length of the sequence containing the encrypted message. The triangular sets are "short" = (0, 0, 24), "middle" = (40, 16, 64) and "long" = (80, 56, 80).

The three input dimensions are linked through a set of rules based on heuristics to one output dimension [see Additional file 3]. The maximum of each correction code means a cut on the y axis (Figure 8). In the next step the fuzzy controller decides, whether to use an 8/4 Hamming-code, a WDH-code or no mutation correction by using the first-maximum method and recommends it to the user.

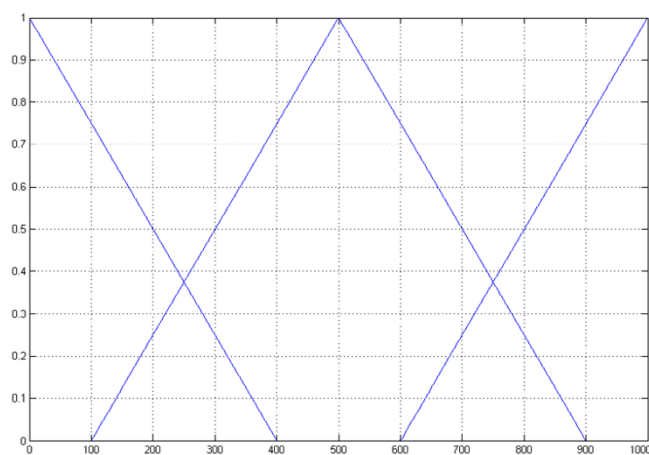


Figure 7

The third input dimension. The third input dimension is the stability over time, which is represented by the number of generations. It is separated into "low" = (0, 0, 400), "middle" = (500, 100, 900) and "high" = (1000, 600, 1000).

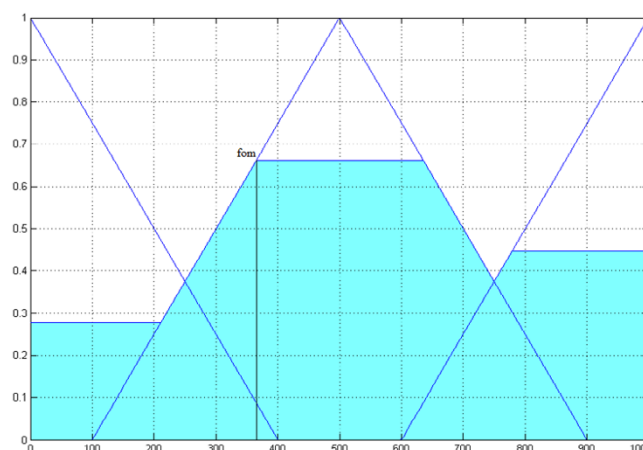


Figure 8

The output dimension of the fuzzy controller. The triangular sets are "none" = (0, 0, 400), "Hamming-code" = (500, 100, 900) and "WDH-code" = (1000, 600, 1000). The maximum of a triangular set, calculated by the set of heuristics of the fuzzy controller, means a cut on the y axis. A cut at 0.28 for none correction code, at 0.67 for Hamming-code and at 0.45 for the WDH-code is shown. The first-of-maximum (fom) represents the recommended correction code, in this case the fuzzy controller recommends the Hamming-code.

Results

The program described above was tested by *in silico* experiments using the DNA sequence encoding the Ypt7 in *Saccharomyces cerevisiae*.

Ypt7

The small GTPases termed Ypt in yeast and Rab in higher eukaryotes are molecular switches in cellular transport processes [21]. Each Ypt protein is localized to the membrane of specific intracellular compartments and highly specific for a particular transport step [22].

The Ypt7 GTPase from *S. cerevisiae* is involved in late endosome-to-vacuole transport and vacuole fusion events [23,24]. Ypt7 is one of the 11 members of the *S. cerevisiae* Ypt family and is homologous to mammalian Rab7.

Analysis of the Ypt7 DNA sequence showed that 32% of the codons allow synonymous substitutions, resulting in 16 bytes, which could be encrypted (Table 6). The first steganogram contains the message "this is a test" and the second one "yet another test" [see Additional file 4].

The results of the analyses of these steganograms with the fuzzy controller are shown in table 7. Translation with DNA-Crypt and the *Expassy Translate Tool* shows that the translated amino acid sequences are identical [25].

Table 6: DNA-Crypt analysis of the Ypt7 DNA sequence. It is possible to encrypt 16.5 bytes in the Ypt7 DNA sequence.

name	amino acids	syn. codons	bytes
Ypt7	206	32.039%	16.5

The pairwise and the multiple sequence alignments show a few mismatches between the three sequences (Figures 9, 10, 11).

The pairwise sequence alignment was performed with Dotlet and the multiple sequence alignment was performed using ClustalW of the European Bioinformatics Institute with standard settings [26,27].

Discussion

DNA-Crypt produces few sequence mismatches similar to the low noise in image steganography. In case of image steganography one can look at the least significant bits to attack the steganographic algorithms. To attack DNA steganography one can perform pairwise or multiple sequence alignments with the original sequences.

Conclusion

The DNA-Crypt algorithm can encode cryptic messages into DNA sequences, which can be used as watermarks for authentication. DNA-Crypt is a substantial extension to other steganographic algorithms based on DNA, which can be used in combination with a binary encryption algorithm such as AES, RSA or Blowfish and a mutation correction code such as the Hamming-code or the WDH-code. The most appropriate code of these correction codes can be selected by a fuzzy controller, which uses three input dimensions.

Mutations, which cause changes in the reading frame, are problematic and are not appropriate for DNA steganography. Mutations, which change a non-synonymous codon to a synonymous codon or vice versa are more important as these mutations cause errors in the encrypted information. The relevance of these errors depends on the encrypted information. If the encrypted information is an image, e.g. a logo, there would be only a linear colour shift in the image, which is not very relevant and can be corrected very easily. However if the encrypted information

Table 7: Ypt7 steganogram fuzzy controller results

name	ϕ	correction code
steganogram 1	9.48	8/4 Hamming-code
steganogram 2	10.5	8/4 Hamming-code

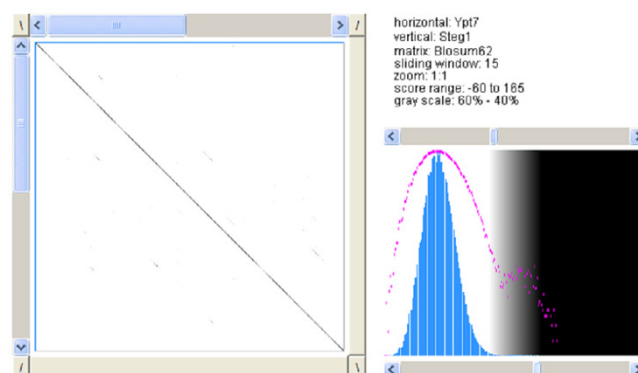


Figure 9
Dotplot of Ypt7 and steganogram 1. Pairwise sequence alignment with Dotlet between the original sequence and the steganogram containing "this is a test" [26].

must remain correct, e.g. a password, the WDH-code must be used to detect these mutations.

We have not encountered any problems so far performing our *in silico* analyses using DNA-Crypt watermarks in DNA coding regions. The use of DNA-Crypt in non-coding sequences like a regulatory RNA sequence or promoter, and enhancer sequences has to be tested *in silico* and *in vivo*. Further analyses to clarify, whether alternative splicing events pose a problem for watermarks still have to be carried out. In conclusion DNA-Crypt algorithm represents an interesting tool for hiding authenticating watermarks within coding DNA sequences *in silico* and most probably in living organisms without affecting the process of protein translation and protein function.

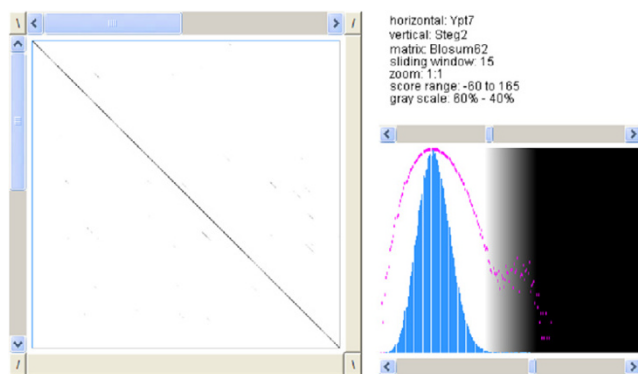


Figure 10
Dotplot of Ypt7 and steganogram 2. Pairwise sequence alignment with Dotlet between the original sequence and the steganogram containing "yet another test" [26].

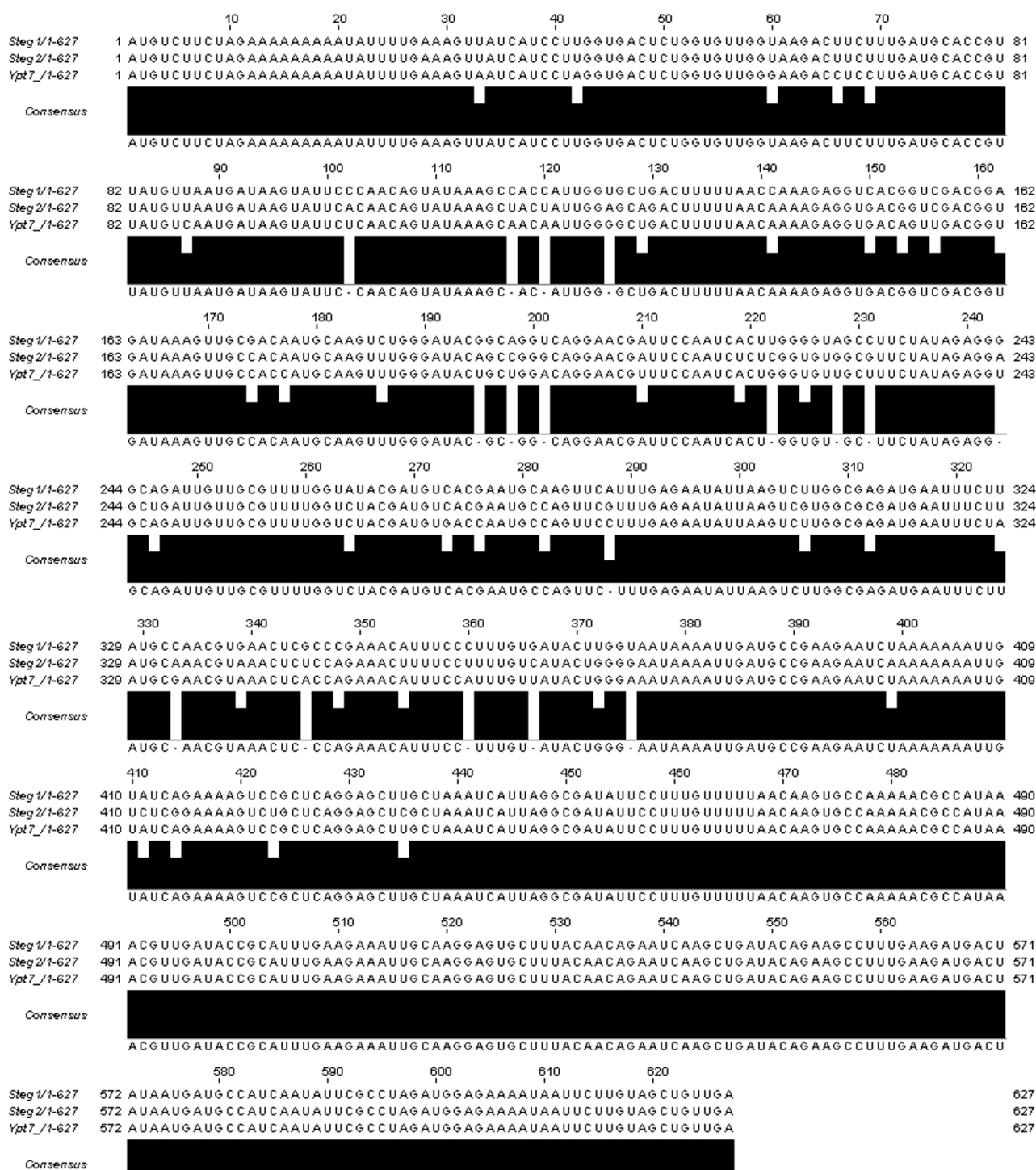


Figure 11
Multiple sequence alignment. Multiple sequence alignment of the original sequence and two steganograms [27].

Availability and requirements

Project Name: DNA-Crypt

Project Homepage: <http://www.uni-muenster.de/Biologie/NeuroVer/Tumorbiologie/DNA-Crypt/index.html>

Operating Systems: Cross-platform

Programming Language: Java 5.0 or higher

Authors' contributions

DH: conception, software development, sequence alignments, figure preparation, manuscript preparation

AB: conception, design, manuscript preparation, coordination, research funds collection. The authors read and approved the final manuscript.

Additional material

Additional file 1

The DNA-Crypt v.2.

Click here for file

[<http://www.biomedcentral.com/content/supplementary/1471-2105-8-176-S1.zip>]

Additional file 2

The fuzzy controller.

Click here for file

[<http://www.biomedcentral.com/content/supplementary/1471-2105-8-176-S2.jar>]

Additional file 3

The set of rules for the fuzzy-controller.

Click here for file

[<http://www.biomedcentral.com/content/supplementary/1471-2105-8-176-S3.txt>]

Additional file 4

The DNA sequences of the Ypt7 and the two steganograms.

Click here for file

[<http://www.biomedcentral.com/content/supplementary/1471-2105-8-176-S4.txt>]

5. Arita M, Ohashi Y: **Secret signatures inside genomic DNA.** *Bio-technol Prog* 2004, **20**:1605-1607.
6. Schneier B: *Applied Cryptography* Pearson Education; 1996.
7. Standards of NI Technology, Eds: **Announcing the ADVANCED ENCRYPTION STANDARD(AES).** *Federal Information Processing Standards Publication* 2001, **197**.
8. Rivest RL, Shamir A, Adleman L, Eds: **On digital signatures and public key cryptosystems.** *MIT Laboratory for Computer Science Technical Memorandum* 1977, **82**.
9. Rivest RL, Shamir A, Adleman L, Eds: *A method for obtaining digital signatures and public-key cryptosystems* New York, NY, USA: Communications of the ACM; 1978.
10. Tanenbaum AS: **The data link layer.** In *Computer Networks* Edited by: Franz M. Prentice Hall.
11. Mamdani EH: **An experiment in linguistic synthesis with a fuzzy logic controller.** *Int. Journal of Man-Machines Studies* 1975, **7**:1-13.
12. Lee CC: **Fuzzy logic in control systems: Fuzzy logic controller.** *IEEE Trans. on Systems Man and Cybernetics* 2000, **20**:404-435.
13. Sugeno M: **An introductory survey of fuzzy control.** *Information Science* 1985, **36**:59-83.
14. Sugeno M, Takagi T: **Fuzzy identification of systems and its applications to modeling and control.** *IEEE Transactions of Systems, Man and Cybernetics* 1985, **15**:116-132.
15. Zadeh LA: **Fuzzy sets.** *Information and Control* 1965, **8**:338-353.
16. Zadeh LA: **A rationale for fuzzy control.** *Journal of Dynamic Systems, Measurement and Control* 1972, **94**:3-4.
17. Zadeh LA: **Outline of a new approach to the analysis of complex systems and decision processes.** *IEEE Transactions of Systems, Man and Cybernetics* 1973, **3**:28-44.
18. Zadeh LA: **The concept of linguistic variable and its application to approximate reasoning, Part 1.** *Information Sciences* 1975, **8**:199-249.
19. Zadeh LA: **The concept of linguistic variable and its application to approximate reasoning, Part 2.** *Information Sciences* 1975, **8**:301-357.
20. Zadeh LA: **The concept of linguistic variable and its application to approximate reasoning, Part 3.** *Information Sciences* 1975, **8**:43-80.
21. Watzke A, Brunsfeld L, Durek T, Alexandrov K, Rak A, Goody R, Waldmann H: **Chemical biology of protein lipidation: semi-synthesis and structure elucidation of prenylated RabGTPases.** *Org Biomol Chem* 2005, **3**:1157-1164.
22. Gotte M, Lazar T, Yoo J, Scheglmann D, Gallwitz D: **The full complement of yeast Ypt/Rab-GTPases and their involvement in exo- and endocytic trafficking.** *Subcell Biochem* 2000, **34**:133-173.
23. Wichmann H, et al.: **Endocytosis in yeast: evidence for the involvement of a small GTP-binding protein (Ypt7p).** *Cell* 1992, **71**:1131-1142.
24. Schimoller F, Riezmann H: **Involvement of Ypt7p, a small GTPase, in traffic from late endosome to the vacuole in yeast.** *J Cell Sci* 1993, **106**:823-830.
25. **ExPASy - Translate tool** [<http://us.expasy.org/tools/dna.html>]
26. **Swiss Institute for Experimental Cancer Research Dotlet** [<http://www.isrec.isb-sib.ch/java/dotlet/Dotlet.html>]
27. **European Bioinformatics Institute ClustalW** [<http://www.ebi.ac.uk>]

Acknowledgements

The authors thank Prof. Dr. Achim Clausing and Dr. Mark Kail for critical reading the manuscript. This work is part of the PhD thesis of DH.

References

1. Clelland C, Risca V, Bancroft C: **Hiding messages in DNA microdots.** *Nature* 1999, **399**:533-534.
2. Gehani A, LaBean TH, Reif JH: **DNA-based cryptography.** *Dimacs Series In Discrete Mathematics and Theoretical Computer Science* 2000, **54**:233-249.
3. Leier A, Richter C, Banzhaf W, Rauhe H: **Cryptography with DNA binary strands.** *BioSystems* 2000, **57**:13-22.
4. Wong PC, Wong KK, Foote H: **Organic data memory using the DNA approach.** *Communications of the ACM* 2003, **46**.

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:
http://www.biomedcentral.com/info/publishing_adv.asp

